

**PRIORITY  
DOCUMENT**

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)



REC'D 04 DEC 2003  
WIPO PCT

**Prioritätsbescheinigung über die Einreichung  
einer Patentanmeldung**

**Aktenzeichen:** 102 49 851.2  
**Anmeldetag:** 25. Oktober 2002  
**Anmelder/Inhaber:** Elektro Beckhoff GmbH,  
Verl/DE  
**Bezeichnung:** Verfahren, Schnittstelleneinheit und Knoten zur  
parallelen Nutzung eines Kommunikationsnetzwerkes  
für Echtzeitanwendungen und Nicht-  
Echtzeitanwendungen  
**IPC:** H 04 L 29/02

**Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ur-  
sprünglichen Unterlagen dieser Patentanmeldung.**

München, den 06. November 2003  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

*[Handwritten signature]*  
Wahl

### Beschreibung

Verfahren, Schnittstelleneinheit und Knoten zur parallelen Nutzung eines Kommunikationsnetzwerkes für Echtzeitanwendungen und Nicht-Echtzeitanwendungen

Das Ethernet ist die am weitesten verbreitete Technologie, mit der in lokalen Kommunikationsnetzen, sog. Local Area Networks (LAN), Daten aktuell mit einer Geschwindigkeit bis zu 100 Mio. Bits/sec. (Mbps) übertragen werden können. LANs sind lokale Kommunikationsnetzwerke, die auf ein geografisches Gebiet begrenzt sind und sich aus einem oder mehreren Servern und Arbeitsstationen, sog. Knoten zusammensetzen, die über ein Kommunikationsleitungsnetz, z.B. ein Koaxial-, Glasfaser- oder Twisted Pair-Kabel verbunden sind. Bei LANs sind verschiedenste Netzwerktopologien möglich, wobei die bekanntesten die Bus-, Ring-, Stern- oder Baumstrukturen sind.

LANs werden mit einem Netzwerk-Betriebssystem und einem einheitlichen Netzwerk-Protokoll betrieben. Das Ethernet stellt ein mögliches Netzwerkprotokoll dar und unterstützt dabei die unterschiedlichsten Kommunikationsprotokolle, z.B. das TCP/IP-Protokoll oder das IPX-Protokoll. Im OSI-Schichtenmodell, dem internationalen Referenzmodell für Datenübertragung in Netzwerken, das aus einem Schichtenstapel aus sieben Schichten aufgebaut ist, wobei für jede Schicht eine Menge von Protokollen definiert ist, die jeweils der nächst höheren Schicht ihre Dienste zur Verfügung stellen, ist das Ethernet der zweiten Schicht, der sog. Leitungsschicht zugeordnet. In dieser Leitungsschicht werden die zu übermittelnden Daten zu Paketen gebündelt, denen spezifische Informationen für das jeweilige Kommunikationsprotokoll hinzugefügt werden. Die Leitungsschicht ist im Netzwerk für den Transport der Datenpakete von Knoten zu Knoten und für die Fehlererkennung zuständig. Beim Ethernet-Konzept ist die Leitungsschicht in zwei Ebenen unterteilt, wobei die erste Ebene den Daten einen Kopfabschnitt, einen sog. Header hinzufügen, der Informatio-

nen enthält, die für eine korrekte Datenübertragung vom Empfängerprotokoll benötigt werden. In der zweiten Ebene des Ethernetprotokolls wird dann das Datenpaket mithilfe eines zusätzlichen Headers und eines weiteren Endabschnitts, eines sog. Trailers, für den Transport der Datenpakete von Knoten zu Knoten eingekapselt. Mit solchen Ethernet-Datenpaketen, den sog. Ethernet-Telegramme, lassen sich Daten mit einer Länge von bis zu 1500 Bytes übertragen.

Ethernet-Protokolle werden vornehmlich bei Bürokommunikationsnetzwerken eingesetzt. Aufgrund der Vorteile des Ethernet-Konzepts bei der Nutzung von Standard-Hard- und -softwarekomponenten sowie der Möglichkeit, bei einfacher Vernetzungs-technologie hohe Datenübertragungsraten zu erreichen, besteht der Wunsch, die Ethernet-Netzwerkkommunikation auch in der industriellen Fertigung zum Datenaustausch zwischen Arbeitsstationen einzusetzen. Insbesondere die mangelhafte Echtzeit-fähigkeit des Ethernet-Netzprotokolls lässt jedoch nur einen begrenzten Einsatz in der Automatisierungstechnik zu. Bei der Steuerung von Maschinen ist es nämlich erforderlich, dass eine zyklische Bearbeitung der Steuerungsaufgabe ohne zeitliche Schwankungen, d.h. mit nur geringen Abweichungen von der gewünschten Zykluszeit im Bereich weniger Mikrosekunden erfolgt, wobei mit einer vorhersehbaren Antwortzeit auf die Regelanforderung reagiert wird.

Echtzeitfähigkeit und schnelle Reaktionszeit, wie sie in der Automatisierungstechnik gefordert werden, haben bei Standard-Datenverarbeitungsanwendungen, bei denen Ethernet-Kommunikation üblicherweise eingesetzt wird, jedoch nur eine unterge-cmte Bedeutung. Zwar sind bereits bei der Video- und Audio-Datenübertragung Ethernet-Kommunikationsverfahren zur Echtzeitübertragung solcher Daten bekannt. Die Anforderung an die Reaktionszeiten sind jedoch bei solchen Anwendungen nicht sehr zeitkritisch, da mit entsprechenden Datenpuffern in der Arbeitsstation kleine Schwankungen ohne weiteres ausgeglichen werden können. Auch in der bidirektionalen Datenübertragung

mittels Ethernet, z.B. bei der Internet-Telefonie oder Online-Gaming spielen die Echtzeitfähigkeit und die Reaktionszeit eine gewisse Rolle.

Um die vorstehenden Anwendungen im Rahmen der Ethernet-Kommunikation möglich zu machen, sind Verfahren zur Priorisierung einzelner Ethernet-Frames entwickelt worden. Hierbei werden vor allem zwei Verfahren eingesetzt. Um eine gewisse Echtzeitfähigkeit mit kurzen Reaktionszeiten zu erreichen, werden die zu versendenden Ethernet-Datentelegramme in verschiedene Prioritätskategorien eingeteilt, wobei zuerst die Datenpakete mit der höchsten Priorität versendet werden. Alternativ werden Verfahren eingesetzt, in denen die Priorität der Daten als zusätzliche Information dem Ethernet-Datenpaket selbst hinzugefügt sind. Nachfolgende Knoten im Netzwerk können anhand dieser zusätzlichen Informationen dann die mit höherer Priorität versehenen Datenpakete ermitteln und diese entsprechend bevorzugt bearbeiten.

Diese beiden vorstehenden Ethernet-Verfahren werden jedoch den hohen Anforderungen bezüglich der Echtzeitfähigkeit, d.h. der synchronen Bearbeitung der Daten mit ihrer Entstehung ohne wahrnehmbare Verzögerung sowie der erforderlichen Reaktionszeiten in der Automatisierungstechnik nicht gerecht. Insbesondere besteht das Problem, dass selbst bei einer optimalen Zuordnung von Prioritäten zu den Ethernet-Datenpaketen ein gerade abgeschicktes niedrig priorisiertes Ethernet-Telegramm auch dann vollständig versendet wird, wenn gleichzeitig ein neues höher priorisiertes Datenpaket hinzukommt. Das neu hinzukommende, hoch priorisierte Datenpaket wird dann um die Sendezeit des niedrig priorisierten Datenpakets, also bis zu einer Datenlänge eines Ethernet-Datenpaketes von ca. 1.500 Bytes verzögert. Dies kann zu einer zeitlichen Verzögerung bei Steuerungsaufgaben von bis zu ca. 150  $\mu$ s führen, so dass harte Echtzeit-Steuerungsanwendungen mit Zykluszeiten um die 50  $\mu$ s und zulässigen Abweichungen von dieser Zykluszeit von maximal 10  $\mu$ s nicht erfüllt werden können.

Aufgabe der Erfindung ist es, ein Verfahren zum Betreiben eines lokalen Kommunikationsnetzes und eine Schnittstelleneinheit zur Anbindung eines Knotens an ein solches lokales Kommunikationsnetz bereit zu stellen, mit dem sich auf einfache Weise parallel Echtzeitanwendungen und Nicht-Echtzeitanwendungen ausführen lassen.

Diese Aufgabe wird mit einem Verfahren gemäß Anspruch 1, einer Schnittstelleneinheit gemäß Anspruch 8 und einem Knoten gemäß Anspruch 13 gelöst. Bevorzugte Weiterbildungen sind in den abhängigen Ansprüchen angegeben.

Gemäß der Erfindung wird in einem Kommunikationsnetz mit mehreren Knoten, die über einen Kommunikationsweg miteinander verbunden sind, eine Datenübermittlung zyklisch und deterministisch ausgeführt, wobei Daten für Echtzeitanwendungen priorisiert behandelt werden, so dass in einem Übertragungszyklus zuerst alle Daten für Echtzeitanwendungen übergeben werden und in der Zeit, die bis zum nächsten Übermittlungszyklus verbleibt, dann die Daten für Nicht-Echtzeitanwendungen.

Das erfindungsgemäße Datenübertragungsverfahren für ein lokales Kommunikationsnetz ermöglicht die parallele Nutzung des Kommunikationsnetzes für Echtzeitanwendungen und Nicht-Echtzeitanwendungen. Insbesondere wird gewährleistet, dass das erfindungsgemäßen Protokoll für lokale Kommunikationsnetze, das die Leitungsschicht im OSI-Modell darstellt, auch den hohen Anforderungen an die Echtzeitfähigkeit und Reaktionszeit für Maschinensteuerungsaufgaben genügt. Um die Kommunikationsanforderungen eines Echtzeitssystems zu erfüllen, erfolgt die Datenübermittlung unter der vollständigen Kontrolle des Echtzeitssystems, wobei die Echtzeitdatenkommunikation gegenüber der sonstigen Datenkommunikation, z.B. zur Administration und Fehlerdiagnose, die vom Betriebssystem des Knotens initiiert wird, priorisiert ist. Alle Daten für Nicht-Echtzeitanwendungen werden nämlich grundsätzlich als nieder prio-

risiert behandelt und erst dann übermittelt, wenn die Datenübermittlung für Echtzeitdatenpakete bereits abgeschlossen ist. Mithilfe der erfindungsgemäßen Priorisierungstechnik für Echtzeitdaten bei gleichzeitiger Ausführung einer zyklischen und deterministischen Datenübermittlung wird dafür gesorgt, dass nicht echtzeitfähige Zugriffe grundsätzlich erst nach den echtzeitfähigen Zugriffen erfolgen und damit den Echtzeitdatenverkehr nicht behindern.

Gemäß einer bevorzugten Ausführungsform wird ein Sendevorgang so ausgeführt, dass bei einem Sendezyklus die Daten für Echtzeitanwendungen erst vollständig gesendet und dann die Zeit berechnet wird, die noch bis zum nächsten Sendezyklus verbleibt, um dann die verbleibende Zeit zur Datenübertragung für Nicht-Echtzeitanwendungen zu verwenden. Es wird so sicher gestellt, dass Datenpakete aus dem Echtzeitsystem immer eine freie Sendeleitungen vorfinden, wenn sie an der Reihe sind. Datenpakete für Nicht-Echtzeitanwendungen, z.B. des Betriebssystems, werden erst danach in den Lücken zwischen zwei Sendezyklen verschickt, wenn entsprechende Zeit verbleibt. Bevorzugt ist dabei, die Daten in Form von Datenpaketen so zu übermitteln, dass dann, wenn die nach dem Senden der Daten für Echtzeitanwendungen verbleibende Zeit die zum Versenden eines Datenpaketes für Nicht-Echtzeitanwendungen benötigte Zeitdauer übersteigt, das entsprechende Datenpaket zwischen gespeichert und bevorzugt mit dem nächsten Sendezyklus versandt wird. Diese Vorgehensweise ermöglicht eine einfache Prioritätssteuerung des Sendebetriebs von Echtzeitanwendungen und Nicht-Echtzeitanwendungen.

Gemäß einer weiteren bevorzugten Ausführungsform werden bei einem Empfangsvorgang die in einem Empfangszyklus empfangenen Daten ausgewertet, um zu bestimmen, welche der empfangenen Daten Daten für Echtzeitanwendungen und welche der empfangenen Daten Daten für Nicht-Echtzeitanwendungen darstellen, wobei die empfangenen Daten für Echtzeitanwendungen im darauf folgenden Empfangszyklus bearbeitet werden. Diese Vorgehens-

weise ermöglicht es, auf einfache Weise alle echtzeitrelevanten Datenpakete herauszufiltern, ohne dass es darauf ankommt, wann bzw. in welcher Reihenfolge sie eintreffen. So wird sichergestellt, dass die echtzeitrelevanten Informationen komplett vor Beginn des nächsten Empfangszyklus vollständig für die auszuführenden Echtzeitanwendungen bereit stehen.

Bevorzugt ist dabei weiterhin, die herausgefilterten empfangenen Daten für Nicht-Echtzeitanwendungen in einem von den Echtzeitanwendungen unabhängigen Vorgang zu bearbeiten. Die Übergabe der Datenpakete mit nicht echtzeitrelevanten Daten außerhalb des Echtzeit-Kontextes an das Betriebssystem des Knotens ermöglicht eine hohe Datenübertragungsrate bei gleichzeitiger einfacher Implementierung des Empfangsvorgangs.

Gemäß einer weiteren bevorzugten Ausführungsform wird ein paralleles Senden, Empfangen und Verarbeiten der Daten für Echtzeitanwendungen und der Daten für Nicht-Echtzeitanwendungen durchgeführt, wobei zum Verarbeiten der Daten in einem ersten Schritt die in einem vorangegangenen Empfangszyklus empfangenen Daten ausgewertet werden, in einem zweiten Schritt die Echtzeitanwendungen mit den ermittelten echtzeitrelevanten Daten ausführt werden und in einem dritten Schritt die zu sendenden Echtzeitdaten dann übergeben werden. Diese Vorgehensweise ermöglicht eine hohe Übertragungsleistung bei vollständiger Echtzeitfähigkeit und gewährleistet somit eine Kommunikationsanforderung für eine schnelle Maschinensteuerung. Bevorzugt ist dabei, den Sende- und Empfangszyklus im Full-Duplex-Betrieb so auszulegen, dass die Zyklusdauer fest ist, der Sendezyklus gegenüber dem Empfangszyklus jedoch um eine konstante Zeitspanne verzögert ist, die die Zeitdauer des ersten und zweiten Verarbeitungsschrittes, bei dem die Daten ausgewertet und die Echtzeitanwendungen durchgeführt werden, entspricht. Diese Vorgehensweise sorgt für einen einfachen und schnellen Full-Duplex-Betrieb mit reibungsloser Echtzeitkommunikation.

Die Erfindung wird anhand der beigefügten Zeichnungen näher erläutert. Es zeigen:

Fig. 1 mögliche Netzwerk-Topologien für ein lokales Kommunikationsnetz;

Fig. 2 einen möglichen Anschluss eines Knoten an ein Netzwerk im Ethernet;

Fig. 3 eine herkömmliche Auslegung eines Netzsystems mit einem getrennten Ethernet-Controller für Nicht-Echtzeitanwendung und einem Feldbussystem für Echtzeitanwendungen;

Fig. 4 einen erfindungsgemäßen Treiber zur parallelen Ausführung von Echtzeitanwendungen und Nicht-Echtzeitanwendungen auf einem Ethernet-Controller;

Fig. 5 einen erfindungsgemäßen Sendevorgang von Echtzeit- und Nicht-Echtzeit-Datenpaketen; und

Fig. 6 einen erfindungsgemäßen Full-Duplex-Betrieb zum parallelen Versenden, Empfangen und Verarbeiten von Echtzeit-Datenpaketen und Nicht-Echtzeit-Datenpaketen.

Mit einem Kommunikationsnetzwerk lassen sich auf einfache Weise Daten und Ressourcen zwischen Arbeitsstationen, im Allgemeinen Computer oder Maschinen, im weiteren auch Knoten genannt, austauschen und gemeinsam nutzen. Das Ethernet-Konzept ist dabei der am weitesten verbreitete Kommunikationsstandard in lokal begrenzten Kommunikationsnetzwerken (LAN). Das Ethernet basiert dabei auf einem LAN-Aufbau, dem eine Mehrzahl von Knoten über ein gemeinsames Übertragungsmedium miteinander verbunden sind, wobei das Ethernet-Konzept die Verkapslung der zu übermittelnden Daten in sog. Datenpaketen, im weiteren auch als Telegramm bezeichnet, mit vorbestimmtem Format vornimmt. Das Ethernet besteht aus drei Bereichen,

nämlich, dem Übertragungsmedium und den Netzwerkschnittstellen, also der Hardware, der Menge von Protokollen, die den Zugriff auf ein Übertragungsmedium steuern und der Ethernet-Paketformat. Das Ethernet stellt grundsätzlich ein Busnetz dar, wobei beliebigen Netztopologien, wie sie in Figur 1 gezeigt sind, genutzt werden können.

Figur 1a zeigt eine weitere mögliche Netztopologie für ein Ethernet. Hier sind die Knoten 1 sternförmig über Punkt-zu-Punkt-Verbindungen 2 mit einer zentralen Vermittlungsstelle 3, einem sog. Switcher, verbunden. Figur 1b stellt als mögliche Netztopologie ein Bussystem dar. Hier sind alle Knoten 1 über eine Übertragungsleitung 2 miteinander verbunden. In Figur 1c eine weitere Ethernet-Netztopologie in Form eines Baumnetzes dar, das eine Kombination der vorstehenden Netzsysteme ist, die mittels zwischengeschalteten Vermittlungsstationen 4, sog. Hubs, verbunden sind.

Figur 2 zeigt einen möglichen Anschluss eines Knotens 1 in einer Netzwerktopologie mit Punkt zu Punkt-Verbindungen. Beim Ethernet erfolgt der Zugriff auf das Übertragungsmedium 2 im Allgemeinen über eine Kopplereinheit 10, einen sog. Transceiver, der in der Regel direkt auf einer Netzwerkadapterkarte zusammen mit einem Ethernet-Controller 12 angeordnet ist. Der Ethernet-Koppler 10 ist dabei über ein Interface 11 und dem Ethernet-Controller 12 verbunden. Alternativ kann der Ethernet-Koppler auch in den Ethernet-Controller integriert sein. Der Ethernet-Controller nimmt die Kodierung der zu sendenden Daten und die Dekodierung der empfangenen Daten vor. Weiterhin werden durch den Ethernet-Controller 12 auch alle zum Betrieb des Ethernet notwendigen Steuermaßnahmen verwaltet, d.h., der Ethernet-Controller führt das Management und die Paketbildung durch Verkapselung der Daten durch. Neben diesem Link-Management dient der Ethernet-Controller 12 auch als Schnittstelle zur Datenstation, dem Knoten, in dem die Daten dann weiterverarbeitet werden.

Die Ethernet-Datenübertragung findet dabei üblicherweise mit Hilfe eines Netzwerk-Protokolls so statt, dass eine Datenübertragung nur durchgeführt wird, wenn das Netz ruhig ist. Darüber hinaus ist zusätzlich eine Kollisionsverhinderungsmechanismus vorgesehen. Ein Ethernet-Datenpaket kann ca. 1.500 Bytes enthalten, wobei die Daten mittels Header- und Trailer-Informationen, die die Anfangskennzeichnung, die Ziel- und Quelladresse, den Datenpakettyp und den Fehlererkennungsmechanismus angeben, ver kapselt sind.

Die Betriebssysteme der an der Netzkommunikation beteiligten Knoten weisen in der Regel eine geschichtete Softwarestruktur auf, um eine protokollspezifische Bearbeitung von einer telegramm- und hardware spezifische Bearbeitung zu trennen. Dadurch ist es möglich, unterschiedliche Kommunikationsprotokolle beim Ethernetstandard einzusetzen, ohne jeweils am hardware spezifischen Treiber Änderungen durchführen zu müssen. Gleichzeitig besteht aber auch die Möglichkeit, im Rahmen des Ethernetstandards die Hardware zu ändern, ohne protokollspezifische Softwareänderungen ausführen zu müssen. Das eingesetzte Kommunikationsprotokoll kann bestimmen, an welchen Ethernet-Controller ein Ethernet-Telegramm übergeben wird. Die vom Ethernet-Controller empfangene Telegramme werden wiederum allen Protokollen des Betriebssystems im Knoten zur Verfügung gestellt, wobei die Protokolle entscheiden, ob sie das Telegramm dann bearbeiten.

Das Ethernet hat sich vor allem als Kommunikationsstandard für Netzwerksysteme in der Bürokommunikation durchgesetzt, da Standard-Hardwarekomponenten und Standard-Softwareprotokolle genutzt werden können und hohe Datenübertragungsraten möglich sind. Aus diesem Grund ist es auch wünschenswert, den Ethernetstandard in industrieller Umgebung nutzen zu können. Das wesentliche Problem besteht hierbei in der mangelnden Echtzeitfähigkeit, so dass bei Automatisierungsaufgaben, wie in Figur 3 gezeigt, herkömmlicherweise die Netzwerksysteme zur Ausführung von nicht zeitkritischen Kommunikationsaufgaben

und von Echtzeitanwendungen zur Maschinensteuerung getrennt sind. Für die Echtzeitanwendungen sind in der Regel Kommunikationsnetze mit eigenständigen Steuerungsbaugruppen, sog. Feldbus-Controllern, in der Arbeitsstation vorgesehen, um zeitkritische Steuerungsaufgaben durchführen zu können. Mit solchen Feldbusssystemen für Echtzeitanwendungen lassen sich herkömmlicherweise Zykluszeiten zur Maschinensteuerung von 50  $\mu$ s bei zulässigen Jitterzeiten, d.h. Abweichungen von der gewünschten Zykluszeit um 10  $\mu$ s erreichen.

Um lokale Kommunikationsnetze für nicht-zeitkritische Standardanwendungen, insbesondere nach dem Ethernetstandard, auch als echtzeitfähiges Datenübertragungsnetz zur Ausführung von Steuerungsaufgaben nutzen zu können, wird, wie in Figur 4 dargestellt, der hardware- und/oder softwarespezifische Treiber herkömmlicher Ethernet-Controller durch einen erfindungsgemäßen und für den Echtzeitbetrieb erweiterten Ethernet-Treiber ersetzt. Dieser erfindungsgemäßen Ethernet-Controller ermöglicht die parallele Nutzung des Kommunikationsnetzes zur Übermittlung von Daten für Echtzeitanwendungen und von Daten für Nicht-Echtzeitanwendungen. Die Datenübermittlung wird dabei zyklisch und deterministisch ausgeführt, wobei die Daten für Echtzeitanwendungen priorisiert behandelt werden, so dass in einem Übermittlungszyklus zuerst alle Echtzeitdaten übergeben werden und in der Zeit, die noch bis zum nächsten Übermittlungszyklus bleibt, dann die Daten für Nicht-Echtzeitanwendung. Alle in Form von Telegrammen zusammengefassten Daten für Nicht-Echtzeitanwendungen, insbesondere die von dem Betriebssystems des dem Ethernet-Controller zugeordneten Knotens erzeugten Telegramme, werden als niedrig priorisiert bezeichnet und sind dem Echtzeitsystem untergeordnet. Der erfindungsgemäße Ethernet-Controller kommuniziert mit dem Echtzeitsystem und ist verantwortlich für das echtzeitfähige Senden und Empfangen der Ethernettelegramme. Um den parallelen Betrieb von Standardanwendungen zu ermöglichen, meldet sich der erfindungsgemäße Ethernet-Controller zusätzlich beim Betriebssystem des Knotens als Standard-Ethernettreiber an,

dem alle Kommunikationsprotokolle des Betriebssystems prinzipiell zur Verfügung stehen. Alle Daten des Betriebssystems werden von erfindungsgemäßen Ethernet-Controller nachrangig behandelt und erst nach einer Kontrolle durch das Echtzeit-System versandt. Alle empfangenen Datentelegramme werden durch das Echtzeit-System begutachtet und dann gffs. an das Betriebssystem weitergereicht.

Der erfindungsgemäße Ethernet-Controller befindet sich also vollständig unter Kontrolle des Echtzeit-Systems, so dass nicht echtzeitfähige Zugriffe verhindert bzw. in echtzeitkritische Bereiche verlagert werden. Das priorisierte Versenden von Echtzeit- und Standardtelegrammen erfolgt dabei in der in Figur 5 gezeigten Weise. Beim Sendevorgang wird in einem Sendezyklus mithilfe des erfindungsgemäßen Ethernet-Controllers sichergestellt, dass die Daten für die Echtzeitanwendung vollständig gesendet und dann erst die Daten für die Nicht-Echtzeitanwendung übermittelt werden. Durch den zyklisch und deterministisch ausgeführten Sendevorgang wird gewährleistet, dass zu versendende Echtzeittelegramme ohne Verzögerung gesendet werden. Gleichzeitig wird die Länge der zu versendenden Echtzeittelegramme aufaddiert, um die Sendedauer bei bekannter Datenrate festzustellen. Nachdem alle Echtzeittelegramme versendet sind, wird die Zeit berechnet, die noch bis zum nächsten Sendezyklus verbleibt. In dieser verbleibenden Zeit werden dann die vorhandenen Telegramme mit Nicht-Echtzeitanwendungen, die vom Betriebssystem verschickt werden sollen und vom Treiber des Ethernet-Controllers in einem Puffer, vorzugsweise einem Fifo, abgelegt sind, verschickt. Dabei wird wiederum die benötigte Sendedauer zum Versenden dieser Datentelegramme mit Nicht-Echtzeitanwendungen berücksichtigt. Kann ein Telegramm mit Nicht-Echtzeitdaten nicht mehr in der noch zur Verfügung stehenden Zeit bis zum nächsten Sendezyklus versandt werden, so verbleibt es im Fifo-Speicher und wird erst nach dem Versenden der Echtzeittelegramme im nächsten Sendezyklus versandt.

Beim Empfangsvorgang werden mithilfe des erfindungsgemäßen Ethernet-Controllers die in einem Empfangszyklus empfangenen Daten ausgewertet, um zu bestimmen, welche der empfangenen Daten Daten für Echtzeitanwendungen und welche der empfangenen Daten Daten für Nicht-Echtzeitanwendungen sind, wobei die empfangenen Daten für Echtzeitanwendungen dann zu Beginn der nächsten Echtzeit bearbeitet werden. Die Daten für Nicht-Echtzeitanwendungen werden dagegen in einem von der Echtzeit-anwendung unabhängigen Vorgang bearbeitet.

Die bevorzugte Vorgehensweise ist hierbei, dass die empfangenen Ethernettelegramme in ihrer Empfangsreihenfolge vom erfindungsgemäßen Ethernet-Controller vorzugsweise im Speicher des Knotens abgelegt werden. Der Ethernet-Controller ist dabei so eingestellt, dass er keine Interrupts beim Empfang auslöst, so dass alle empfangene Telegramme erst am Anfang des nächsten Echtzeitzyklus bearbeitet werden. Dabei geht dann das Echtzeitsystem alle empfangenen Telegramme durch und wertet die für das Echtzeitsystem relevanten Telegramme aus. Alle empfangenen echtzeitrelevanten Daten stehen somit zu Beginn des Echtzeitzyklus zur Verfügung. Alle Telegramme mit nicht-echtzeitrelevanten Daten werden dagegen als gelesen markiert und außerhalb des Echtzeit-Kontextes an das Betriebssystem übergeben. In einem von der Echtzeitanwendung unabhängigen Prozess kann dann die Liste der empfangenen markierten Telegramme vom Betriebssystem und dessen Protokollen bearbeitet werden. Die vom Echtzeitsystem bearbeitete Echtzeittelegramme werden dagegen in der Regel nicht an die Protokolle des Betriebssystems weitergegeben. Eine solche Weitergabe erfolgt nur zu Debug-Zwecken, um ggfs. Fehler bei der Echtzeitanwendung festzustellen.

Bevorzugt ist weiterhin ein Ethernets Einsatz für Echtzeit- und Nicht-Echtzeitanwendungen im Full-Duplex-Betrieb, d.h. die Möglichkeit, ein paralleles Senden, Empfangen und Verarbeiten von Daten für Echtzeitanwendungen und für Nicht-Echtzeitanwendungen durchzuführen. Ein erfindungsgemäßer Full-Duplex-

Betrieb ist in Figur 6 gezeigt. Es sind dabei mehrere aufeinanderfolgende Echtzeitzyklen dargestellt, wobei jeweils der Zeitablauf der Verarbeitung durch die zentrale Verarbeitungseinheit (CPU) im Knoten sowie der Ablauf der Sende- und Empfangsvorgänge von Echtzeittelegrammen und Standardtelegrammen mit Nicht-Echtzeitdaten dargestellt ist. Im Schritt 1 werden jeweils von der zentralen Verarbeitungseinheit zu Beginn eines Echtzeitzyklus die empfangenen Telegramme vom vorhergehenden Übermittlungszyklus ausgewertet. Im Schritt 2 wird dann auf der Grundlage der ausgewerteten Echtzeitdaten die eigentliche Automatisierungsaufgabe in Echtzeit durchgeführt. Anschließend wird von der zentralen Verarbeitungseinheit im Schritt 3 die zu versendenden Echtzeittelegramme an den Ethernet-Controller übergeben. Hierbei werden dann immer zuerst sofort die echtzeitrelevanten Telegramme übermittelt und dann - falls vorhanden und ausreichende Zeit verbleibt - die Telegramme mit nicht-echtzeit relevanten Daten des Betriebssystems. Der Sendevorgang ist dabei, bezogen auf den Beginn eines Echtzeitzyklus, um eine feste Zeitspanne verzögert, die der Zeitdauer für den ersten und zweiten Bearbeitungsschritt der CPU entspricht. Der Empfangsvorgang dagegen wird vorzugsweise gleichzeitig oder kurz nach dem Beginn des Echtzeitzyklus gestartet.

Durch die erfindungsgemäße Erweiterung des Ethernetkonzepts ist es möglich, eine parallele Nutzung von lokalen Netzwerken für Standardanwendungen und Echtzeitanwendungen vorzunehmen.

## Patentansprüche

1. Verfahren zum Übermitteln von Daten für Echtzeitanwendungen und Nicht-Echtzeitanwendungen auf einem Kommunikationsnetzwerk mit mehreren Knoten, die über Kommunikationswege miteinander verbunden sind, wobei die Datenübermittlung zyklisch und deterministisch erfolgt und die Daten für Echtzeitanwendungen priorisiert behandelt werden, so dass in einem Übermittlungszyklus zuerst alle Daten für Echtzeitanwendungen übergeben werden und in der Zeit, die noch bis zum nächsten Übermittlungszyklus, dann die Daten für Nicht-Echtzeitanwendungen, übergeben werden.
2. Verfahren nach Anspruch 1, wobei beim Sendevorgang in einem Sendezyklus die Daten für Echtzeitanwendungen vollständig gesendet und die Zeit berechnet wird, die noch bis zum nächsten Sendezyklus verbleibt, um dann in der verbleibenden Zeit Daten für Nicht-Echtzeitanwendungen zu senden.
3. Verfahren nach Anspruch 2, wobei die Daten in Form von Datenpaketen übermittelt werden und dann, wenn die nach dem Senden der Daten für Echtzeitanwendungen verbleibende Zeit die zum Versenden eines Datenpakets für Echtzeitanwendungen benötigte Sendedauer übersteigt, das Datenpaket zwischengespeichert und vorzugsweise im nächsten Sendezyklus versandt wird.
4. Verfahren nach einem der Ansprüche 1 bis 3, wobei beim Empfangsvorgang in einem Empfangszyklus empfangenen Daten ausgewertet werden, um zu bestimmen, welche der empfangenen ~~Zeit~~ Daten für Echtzeitanwendungen und welche der empfangenen Daten, Daten für Nicht-Echtzeitanwendungen sind und die empfangenen Daten für Echtzeitanwendungen im nächsten Echtzeitzyklus bearbeitet werden.

5. Verfahren nach Anspruch 4, wobei die empfangenen Daten für Nicht-Echtzeitanwendungen in einem von der Echtzeitanwendung unabhängigen Vorgang bearbeitet werden.

6. Verfahren nach einem der Ansprüche 1 bis 5, wobei ein paralleles Senden, Empfangen und Verarbeiten von Daten für Echtzeitanwendungen und von Daten für Nicht-Echtzeitanwendungen durchgeführt wird und wobei zum Verarbeiten der Daten in einem Echtzeitzyklus in einem ersten Schritt die in einem vorangegangenen Empfangszyklus empfangenen Daten ausgewertet werden, in einem zweiten Schritt die Echtzeitanwendungen ausgeführt werden und in einem dritten Schritt die zusehenden Daten für Echtzeitanwendungen übergeben werden.

7. Verfahren nach Anspruch 6, wobei die Zyklusdauer des Sendezzyklus und des Empfangszyklus dem Echtzeitzyklus entspricht und der Sendezzyklus gegenüber dem Empfangszyklus um eine konstante Zeitspanne verzögert ist, die der Zeitdauer für den ersten und zweiten Verarbeitungsschritt einspricht.

8. Schnittstelleneinheit zur Anbindung eines Knotens an ein Kommunikationsnetzwerk mit mehreren Knoten, die über einen Kommunikationsweg miteinander verbunden sind, wobei die Schnittstelleneinheit zur zyklischen und deterministischen Übermittlung von Daten zwischen dem Knoten und dem Kommunikationsnetzwerk ausgelegt ist, um Daten für Echtzeitanwendungen gegenüber Daten von Nicht-Echtzeitanwendungen priorisiert zu behandeln, so dass in einem Übermittlungszyklus zuerst alle Daten für Echtzeitanwendungen übergeben werden und in der Zeit, die noch bis zum nächsten Zyklus verbleibt, dann die Daten für Nicht-Echtzeitanwendungen übergeben werden.

9. Schnittstelleneinheit nach Anspruch 8 mit einem Sender zum Übermitteln von Daten vom Knoten zum Kommunikationsnetzwerk, um in einem Sendezzyklus die Daten für Echtzeitanwendungen vollständig zu senden und die Zeit zu berechnen, die noch bis zum nächsten Sendezzyklus verbleibt, um dann in der

verbleibenden Zeit Daten für Nicht-Echtzeitanwendungen zu versenden.

10. Schnittstelleneinheit nach Anspruch 9, wobei die Daten in Form von Datenpaketen übermittelt werden und ein Zwischen-speicher vorgesehen ist, um dann, wenn die nach dem Senden der Daten für Echtzeitanwendungen verbleibende Zeit, die zum Versenden eines Datenpaketes für Nicht-Echtzeitanwendungen benötigte Sendedauer übersteigt, das Datenpaket zwischenzuspei-chern und vorzugsweise im nächsten Sendezyklus zu versenden.

11. Schnittstelleneinheit nach einem der Ansprüche 8 bis 10, mit einem Empfänger zum Übermitteln von Daten vom Kommunika-tionsnetzwerk zum Knoten, um beim Empfangsvorgang in einem Empfangszyklus die Daten auszuwerten, um zu bestimmen, welche der empfangenen Daten Daten für Echtzeitanwendungen und wel-che der empfangenen Daten Daten für Nicht-Echtzeitanwendungen sind.

12. Schnittstelleneinheit nach Anspruch 11, wobei die empfangenen Daten für Echtzeitanwendungen und die empfangenen Daten für Nicht-Echtzeitanwendungen unabhängig voneinander bearbeitet werden.

13. Knoten mit einer Schnittstelleneinheit nach einem der Ansprüche 8 bis 12 und mit einer Verarbeitungseinheit, um in einem ersten Schritt, die in einem vorangegangenen Empfangs-zyklus empfangenen Daten auszuwerten, in einem zweiten Schritt die Echtzeitanwendungen mit den empfangenen Daten für die Echtzeitanwendungen auszuführen und in einem dritten Schritt die zu sendenden Daten für Echtzeitanwendungen zu übergeben.

**Zusammenfassung**

Verfahren, Schnittstelleneinheit und Knoten zur parallelen Nutzung eines Kommunikationsnetzwerkes für Echtzeitanwendungen und Nicht-Echtzeitanwendungen

Die Erfindung betrifft eine Technik zum Übermitteln von Daten für Echtzeitanwendungen und Nicht-Echtzeitanwendungen auf einem Kommunikationsnetzwerk mit mehreren Knoten, die über Kommunikationswege miteinander verbunden sind, wobei die Datenübermittlung zyklisch und deterministisch erfolgt und die Daten für Echtzeitanwendungen priorisiert behandelt werden, so dass in einem Übermittlungszyklus zuerst alle Daten für Echtzeitanwendungen übergeben werden und in der Zeit, die noch bis zum nächsten Übermittlungszyklus, dann die Daten für Nicht-Echtzeitanwendungen, übergeben werden.

Figur 6

# Figur für die Zusammenfassung

FIG. 6

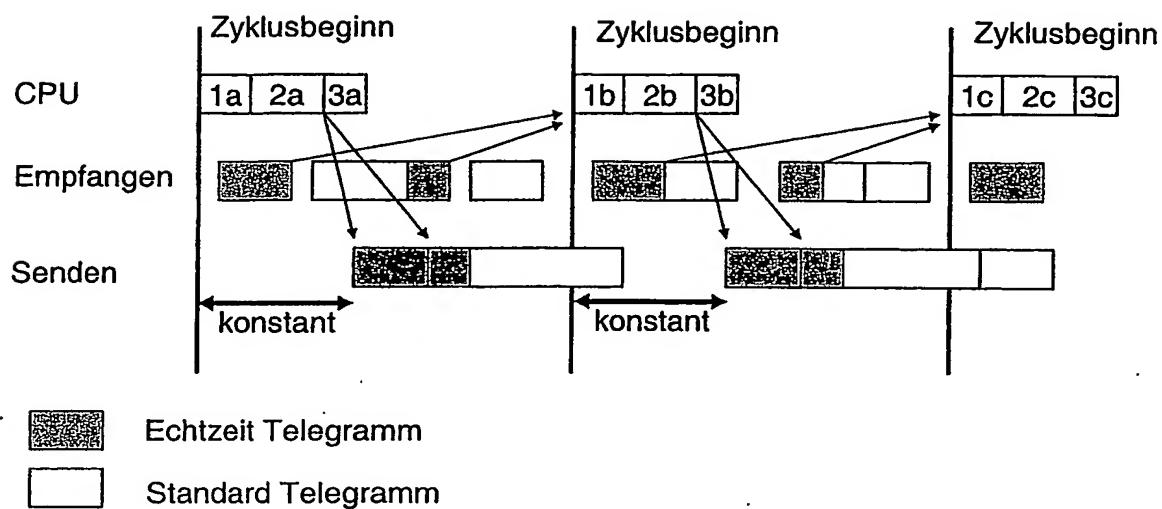


Fig. 1

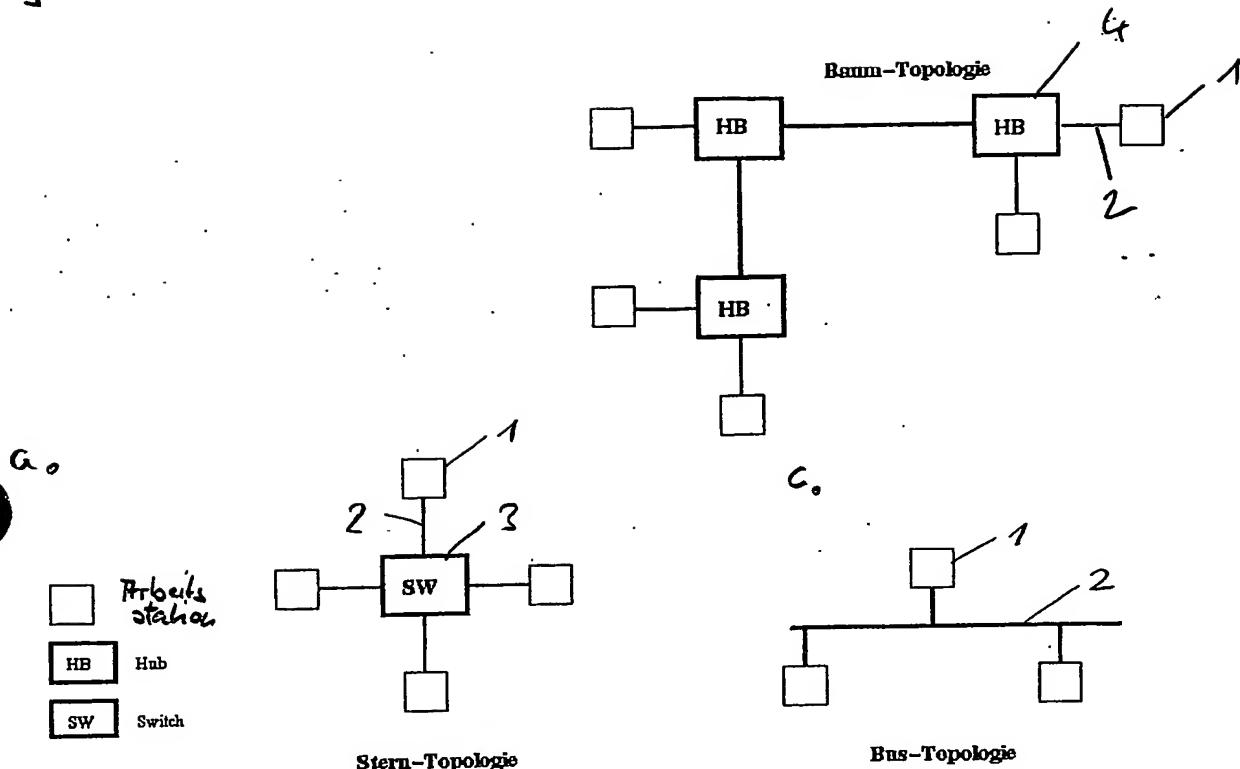
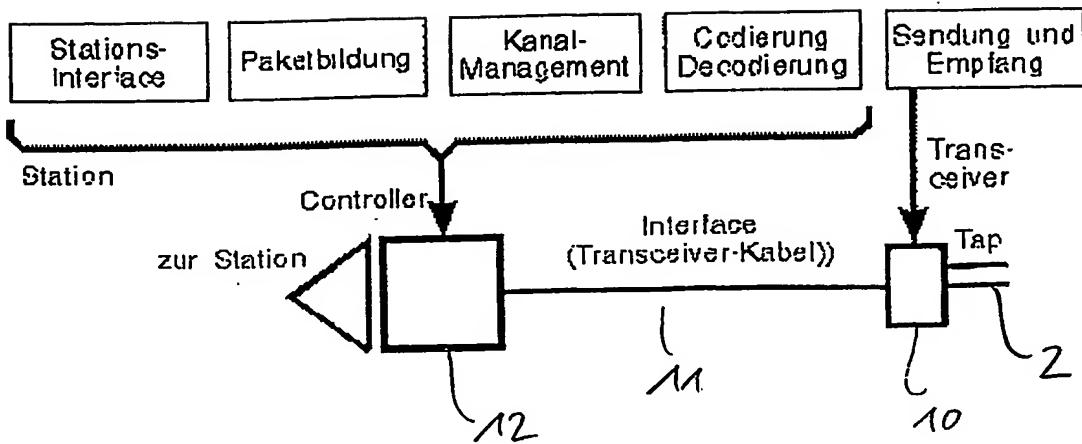


FIG. 2



BEST AVAILABLE COPY

FIG. 3

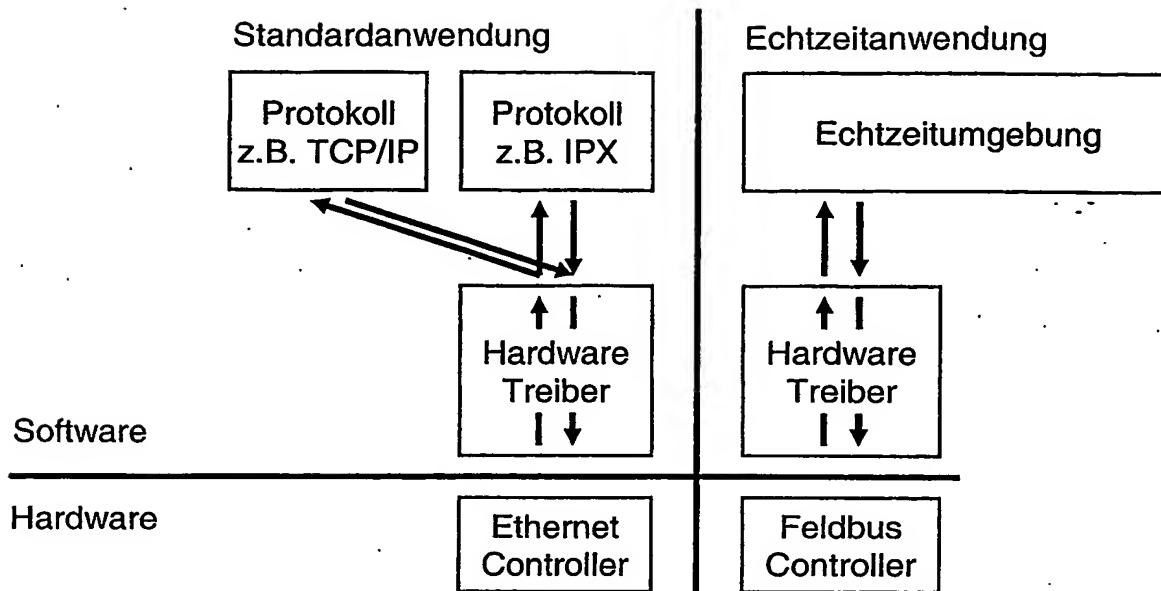


FIG. 4

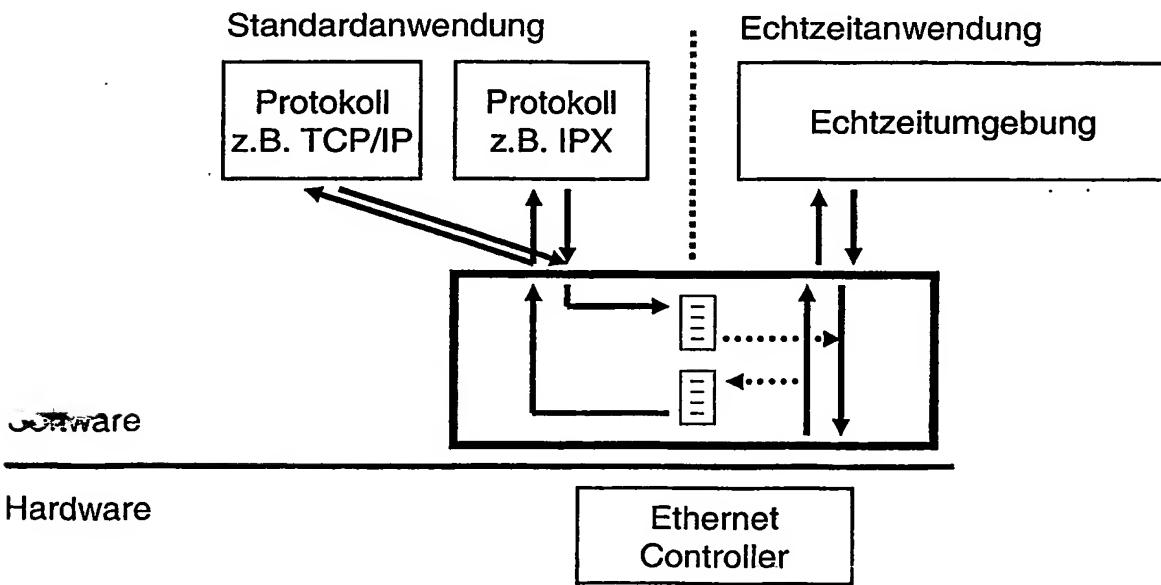


FIG. 5

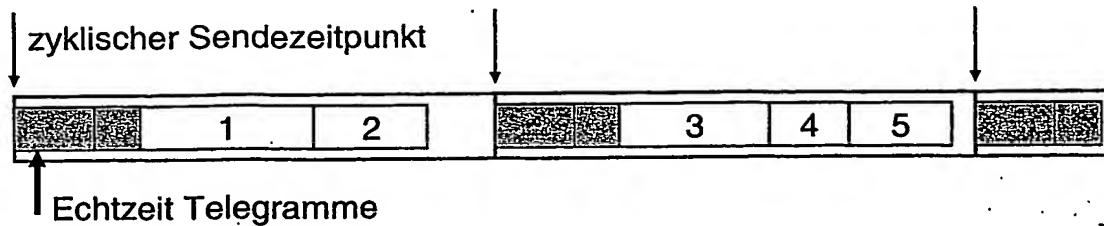
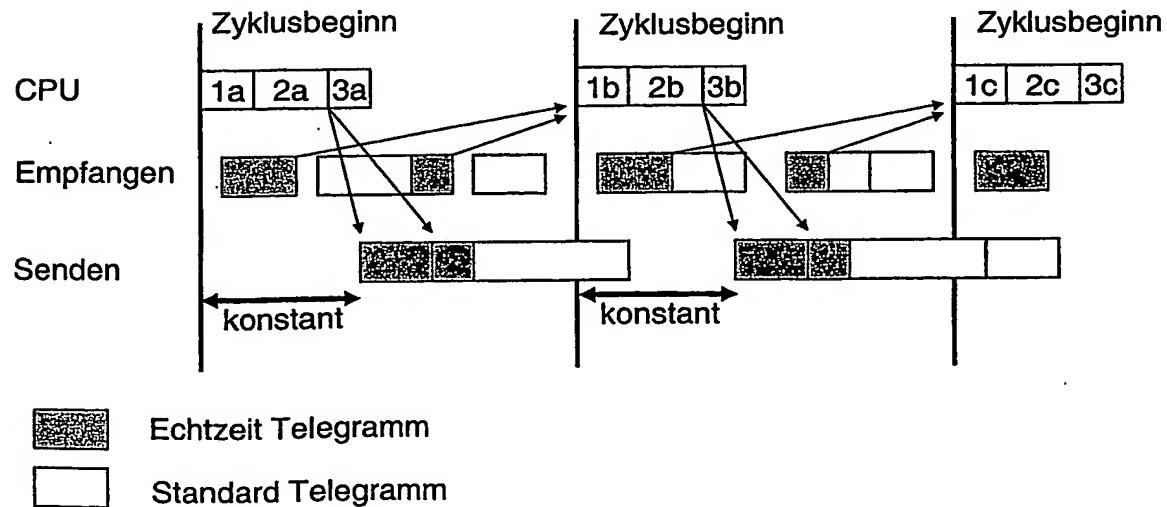


FIG. 6



BEST AVAILABLE COPY